The 2010 IEEE/RSJ International Conference on
Intelligent Robots and Systems
October 18-22, 2010, Taipei, Taiwan

Using On-Line Conditional Random Fields to Determine Human Intent for Peer-To-Peer Human Robot Teaming

John R. Hoare and Lynne E. Parker

Abstract— **In this paper we introduce a system under development to enable humans and robots to collaborate as peers on tasks in a shared physical environment, using only implicit coordination. Our system uses Conditional Random Fields to determine the human's intended goal. We show the effects of using different features to improve accuracy and the time to the correct classification. We compare the performance of the Conditional Random Fields classifiers by testing the classification accuracy with both the full observation sequence, as well as accuracy when the observations are classified as the observations occur. We show that Conditional Random Fields work well for classifying the goal of a human in a box pushing domain where the human can select one of three tasks. We discuss how this research fits into a larger system we are developing for peer-to-peer human robot teams for shared workspace interactions.**

Keywords – **Human-Robot Interaction, Peer-To-Peer Teams, Conditional Random Fields, Human Intent Recognition**

I. Introduction

In peer-to-peer human-robot teaming, a primary objective is to create a style of cooperation between robot(s) and human(s) that is reminiscent of well-practiced human-only teams. In these human-only teams, the individuals have trained together, and understand intuitively how to interact with each other on the current task without the need for any explicit commands or conversations. An example is a fire team of soldiers that have trained together and understand well how to interact with each other to perform a task, such as building clearing. Through practice, these soldiers know various alternative ways to interact, when these various modes of interaction should take place, and with whom. In these applications, the soldier cannot afford to pay any attention to the robot or give commands to the robot. Instead, the robot should implicitly observe the ongoing team actions and respond with appropriate actions to assist the team in achieving its objectives. In this interaction, the human performs tasks in a very natural manner, as he or she would when working with a human teammate; there is no need for PDAs, heads-up-displays, computers, or other types of graphical user interfaces (GUIs) to enable the human to communicate with the robot[1]. In this paper, we focus on the issue of enabling the robot to determine the human's current goals via sensor observation only, without requiring any direct commands from the human. Ultimately, the robot should then respond by helping the human with the task in an appropriate manner, consistent to the inferred human intent; however, this topic is the subject of future research.

Interestingly, the literature does not agree on what exactly is meant by "peer-to-peer" human-robot teaming. Current literature often views humans and robots as peers in terms of decision-making, rather then a team-mate in a shared physical workspace. On the other hand, our research is focused on peer-to-peer teaming with shared workspace interactions. Thus, we define the specific type of peer-to-peer human-robot teaming addressed in this paper as follows:

Peer-to-peer human-robot teams for shared workspace interactions is demonstrated in a system with the following characteristics: humans and robots operate side-by-side in the same physical space, each performing physical actions based upon their skills and capabilities; the team works on a shared cooperative activity, as defined by Bratman [2], in which the agent demonstrates mutual responsiveness, commitment to the joint activity, and commitment to mutual support; and, team members share common ground [7], meaning that they have a shared understanding of the task at hand.

To ground our approach, we are initially developing this system in the context of a human and robot working together to achieve a box-pushing and site clearing task [17]. This benchmark was chosen because it has long served as a canonical multi-robot testbed, offering a clear domain where close coordination and cooperation can be required. However, to our knowledge, no one has used this test domain for demonstrating human-robot interactions in a shared physical workspace. We thus selected this domain because it is a well-understood testbed, and it provides eventual opportunities to compare multi-robot teams with human-robot teams.

In the variant of the box-pushing and site clearing task we use in this research, the robot and human begin with a starting configuration of randomly placed colored boxes, and then move the boxes into a series of goal configurations. The desired goal configuration is determined only by the human through his/her actions; this goal is not explicitly communicated to the robot. The research reported in this paper focuses on enabling the robot to properly infer the human's goal by observing the movement of boxes undertaken by the human.

This paper introduces a method of classifying the human's actions into the desired goal by using Conditional Random Fields (CRFs). In particular, our focus is to maximize classification accuracy while minimizing the length of data sequence that must be seen to correctly classify the human's goal. Additionally, we investigate alternate sets of features

J. R. Hoare and L. E. Parker are with the Distributed Intelligence Laboratory, Department of Electrical Engineering and Computer Science, University of Tennessee, Knoxville, TN 37996-3450, {jhoare, parker}@eecs.utk.edu.

[1]This is not to say that in some future system, we might not want to supplement the information available to the human through some alternative means, such as a heads-up display. However, the focus of our research is to develop technologies that do not require explicit communication devices.

to find the highest performing set. We also compare our classifier to both a simple Decision Tree classifier, as well as human test subjects, to determine the merit of the CRF approach.

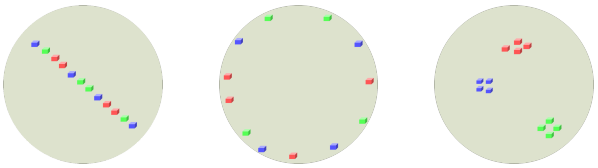
II. Related Work

Human-robot interaction has been studied extensively in the last few years. Of the work that specifically addresses peer-to-peer human-robot teaming (as we define for the context of this paper), most of the work treats the human as a reliable remote source of information or control. Nearly all of this research falls into the topic area of sliding autonomy (also referred to as dynamic autonomy or adjustable autonomy), in which team members can decide if and when to transfer control to another member of the team (e.g., [6], [15]). The different levels of autonomy can be predefined or dynamically defined, and can be changed by a human [5] or autonomously by the robot itself [12]. However, to our knowledge, none of this work on sliding autonomy involves robots working face-to-face with the human, in which both robot and human perform physical skills in a shared workspace.

The work of Reed and Peshkin [14] compares the performance of Human-Human teams with Human-Robot teams in a target acquisition task using a haptic interface. In the task, participants rely only on implicit communication of forces through the crank used to track the target. The authors model their computer to mimic that of a human, and are able to pass the so called "Haptic Turing Test." Similarly to this work, we will base our system for Human-Robot teams on how Human-Human teams function.

Research that involves humans working in the same physical space as the robots, and which is highly relevant to our research objectives, is the work of Hoffman and Breazeal [7]. In this work, the robot is non-mobile, and the workspace is directly between the human and the robot. The human can teach child-like skills to the robot, and then work together to perform tasks (such as pushing buttons or categorizing blocks). Their work makes extensive use of rich physical cues, such as nodding, head poses, hand gestures, facial expressions, eye gaze, shrugging, and so forth. They are able to achieve this research through the use of highly expressive robots. Other work by Hoffman and Braezeal use a human and robot interaction system in a simulated factory setting [8]. Here, the past actions of the human are used to predict the human's most likely future actions, so the robot can obtain the appropriate tool for the task ahead of time.

Our work differs from this prior work in that humans and robots are operating in the same physical space while humans perform normal human actions, using their own bodies. Further, we are not focusing on explicit communication cues from the human (such as specific gestures); instead, we are focusing on creating interactions through implicit communication cues, by having the robot infer intention by analysing normal human actions (and the consequences of those actions) in the current task and environment situation. The specific work reported in this paper focuses primarily on the sensor-based interpretation of the outcome of human actions (e.g., [3], [4]). This issue is commonly referred to as human activity (or intent) recognition. In this paper, we focus on watching the human's effect on the environment (in the form of box motions) to infer the human's intent.



(a) *to-line* task (b) *to-wall* task (c) *to-groups* task

Fig. 1. Examples of goal position of boxes for the three goal types for our task. The circle represents the environment with the border representing a wall.

Fig. 2. Still captures from the overhead cameras during physical experiments at the completion of a *to-group* task.

The approach of our work is highly inspired by the work of Vail, et al. [18]. While [18] uses a CRF for activity recognition in robot-only teams, it is similar to our work in that both attempt to determine intent by using two-dimensional location information. However, the work by Vail uses pre-recorded data and does not make its classification while the data is being observed (which is in contrast to the objective of our research). CRFs have also been used for activity recognition by Liao, et al. [10]. Here, the authors use a multi-layered CRF to determine a human's activity, and then use that activity classification to pick out significant places using GPS data. However, as with the previously mentioned work of Vail, the data used is pre-recorded, and the authors do not attempt to determine these classifications while they are occurring. Our work focuses on making a classification on the sequence stream as it occurs, and therefore before the entire observation sequence has been observed.

III. Approach

A. Introduction

The type of human intent recognition studied in this paper requires recognizing temporal sequences of observations, and classifying them into the most likely category of human intent as the observations are being seen. For our benchmark task domain of box pushing and site clearing, we define three possible objective configurations of the boxes, as follows (and also illustrated in Figure 1):

- *L0*: Boxes all into a single line (*to-line* task)
- *L1*: Boxes pushed to the walls (site clearing) (*to-wall* task)

- *L2*: Boxes pushed into groups of the same color (or that share some other feature) (*to-groups* task).

(Ongoing work is extending this set of goal objectives.)

As the human begins pushing the boxes, the robot must observe the configuration of boxes that evolve over time, and determine which of the goal objectives is most likely. Since raw sensor data contains an extensive amount of information, we make the problem more tractable by defining sets of features that capture the most relevant aspects of the sensory data. Thus, the observational input to the recognizer is a vector of features, calculated from the sensory feedback, that provide relevant data on the human activity, such as the motion of the boxes and the positions of the boxes relative to the walls and each other. The observation sequence X is a set of discrete observations taken from the system where each discrete observation x_i is a vector of features. The entire observation sequence is then defined as $X = \{x_1, ..., x_T\}$. The ground truth label sequence Y then has a corresponding one-to-one mapping with the discrete observations, where $Y = \{y_1, ...y_T\}$; each y_i is either *L0*, *L1*, or *L2*, as defined above. We have defined three primary types of features – Simple box relationships, time-varying "Delta" (Δ) features, and Indicator features. The specific features we have defined are described in detail in Section III-C. One of the objectives of the research reported in this paper is to determine which set of features provides the best learning results.

We have chosen CRFs to determine human intent because they have been shown to perform better than Hidden Markov Models, even when the same assumptions are made [18]. When the independence assumptions of the Hidden Markov Model are broken even better accuracy can be achieved. Alternative techniques include simple classifiers such as decision trees and k-nearest-neighbors. However, we believe the temporal features of our problem will make these simpler classifiers insufficient. For comparison purposes we provide results that contrast the results of the CRF with a simple decision tree classifier.

The learning results are evaluated in terms of two metrics: accuracy and Time To Correct Classification (TTCC). Because our system is to classify human actions as they occur, the CRF must be able to classify using all the information up to the current state in time. To do this, we run the CRF in what we call the "on-line" method. In the "on-line" method, the CRF uses all the observations up to the current time $X = \{x_0, ..., x_t\}$ to classify the label y_t for the current time t. This "on-line" method can run in real time, as the observations occur, given a sufficiently granular sampling rate (our sampling rate of 2Hz is more than enough to allow for "on-line" classification). We compare the results of the system using the "on-line" method to the classical method of using the full observation sequence, which uses the entire observation sequence X to label each observation, x_i with the appropriate label y_i.

B. Conditional Random Fields

As previously noted, the algorithmic learning model that we use for human intent recognition is a Conditional Random Field. CRFs, introduced in [9], provide a framework for building probabilistic models to segment and label sequence data. A CRF is closely related to Hidden Markov Models (HMMs); however, HMMs are generative models, which assign a joint probability to paired observation and label sequences. Such models require an enumeration of all possible observation sequences, which can be impractical if it is desired to represent multiple interacting features, or long-range dependencies of the observations. In contrast, CRFs are discriminative models, which specify the probabilities of possible label sequences given an observation sequence; thus, modeling effort is not expended on the observations. CRFs also have the advantage of containing any number of feature functions, which can examine the entire input sequence at any point during inference.

CRFs are represented as undirected graphs, in which vertices represent random variables whose distribution is to be learned, and edges represent possible dependencies between random variables. The input sequence of observations is named $X = \{x_1, x_2, ..., x_T\}$, while $Y = \{y_1, y_2, ..., y_T\}$ represents the labels (i.e., unknown state variables) that need to be inferred given the observations. In a CRF, discrete random variables, Y, are conditionally dependent on an input sequence X. The layout of the graph of random variables is a chain, with an edge between each y_{t-1} and y_t. CRFs represent the conditional probability, $P(Y|X)$, as the product of potential functions that are computed over each clique in the graph. Each potential function is computed in terms of feature functions on the observations and adjacent pairs of labels. Thus, y_i's conditional dependency on X is defined through a fixed set of *feature functions* of the form $f(i, y_{t-1}, y_t, X)$. These feature functions, applied to the input sequence, help determine the likelihood of each possible value for y_t. Each feature is associated with a numerical weight; multiple features are then combined to determine the probability of a certain value for y_t. For space purposes, we omit the detailed probabilistic formulation of CRFs; more details are available in [9].

Training data provides the label of the current activity for each observation. The CRF is then trained to learn the conditional distributions between the y_t values given the input x_t. Conditional Random Fields are typically trained by estimating a parameter vector that maximizes the conditional log-likelihood of the training data. Standard optimization techniques such as limited memory Broyden-Fletcher-Goldfarb-Shanno [11] are typically applied, and have proven to be very fast [19]. The learned model is then used on-line to infer the most likely (Viterbi) labeling, $y^* = argmax_y p(y|x)$ using an inference algorithm such as the forward-backward algorithm [16] which runs in $O(M^2T)$ time [16] where M is the number of states, and T is the number of observations in the sequence. Details on the inference algorithm for CRFs are given in [16].

C. Description of Features

This section introduces and defines the set of features that the Conditional Random Fields classifier uses to make

classifications. First we define a set of "simple" features:

- $f_1(S)$ = *Error to a line fit*. This feature is calculated using a least-squares line fit to the set of boxes in S, and then calculating the average distance away from that line.
- $f_2(S)$ = *Average distance to the wall*.
- $f_3(S)$ = *Average distance to the centroid*.

We apply features f_1 and f_2 to the set of all boxes. For feature f_3 we generate three features by applying f_3 to each color set of boxes. This gives us the following set of features:

$$f_{\text{simple}}(t,S) = \{\ f_1(S[t]),\ f_2(S[t]),\\ f_3(S_{\text{R}}[t]),\ f_3(S_{\text{G}}[t]),\ f_3(S_{\text{B}}[t])\ \}$$

where $S[t]$, $S_R[t]$, $S_G[t]$, and $S_B[t]$ are the set of all boxes, the set of red boxes, the set of green boxes, and the set of blue boxes, respectively, at time t.

We next add the concept of a Δ feature, which is a feature calculated by the change over time (w) of another feature. The Δ features are analogous to the rate of change of the Simple features. To calculate these features, we subtract the value of the current feature from the value of that feature a time period w ago. One Δ feature is calculated for each Simple feature and takes the form of:

$$f_\Delta(t,w,S) = \{f_i(S[t]) - f_i(S[t-w])\}, \forall\ f_i \in f_{\text{simple}}$$

Finally, we define several binary Indicator features that provide task-specific information. These are defined generally as:

$$I(x) = \begin{cases} 0 & \text{if } x \text{ is False} \\ 1 & \text{if } x \text{ is True} \end{cases}$$

Many of the Indicator features indicate the status of the last moved boxes. Because of this, we must define the following notation. We define a list of boxes:

$$B_1, B_2, B_3, ..., B_p\ \forall p \in \text{Pushed-Boxes}$$

where B_p is the most recently pushed box, B_{p-1} is the second to most recently pushed box, and so on. We define a pushed box to be one that has been moving and has come to rest. Note that this does not have a one-to-one mapping to real boxes, as one box could be B_1 and then B_5 if that box was pushed first, and then pushed after four other boxes were pushed.

The Indicator features are defined as follows:

- $f_4(S,b) = I(b$ boxes within S are within threshold ϵ_{group} of each other$)$
- $f_5(B_p) = I(p^{th}$ last moved box was moved to within threshold ϵ_{wall} of the wall$)$
- $f_6(S,B_p) = I(p^{th}$ last moved box was moved within threshold ϵ_{group} of like-colored boxes in $S)$
- $f_7(\{B_p, B_{p-1}, \ldots, B_{p-b}\}) = I($Last b moved boxes were pushed within threshold ϵ_{line} of a line$)$

We use these Indicator features to create a larger set of Indicator features for different values of b and different sets of boxes. First, we use Indicator feature f_4 to indicate whether 2, 3, and 4 boxes for each color of boxes are clustered together. Because we have three colors of boxes this set of features is defined as:

$$f^1_{\text{indicator}}(t,S) = \{f_4(S_R[t],2),\ f_4(S_R[t],3),\ \ f_4(S_R[t],4),\\ f_4(S_G[t],2),\ f_4(S_G[t],3),\ \ f_4(S_G[t],4),\\ f_4(S_B[t],2),\ f_4(S_B[t],3),\ \ f_4(S_B[t],4)\}$$

To be indicative of the *to-wall* task, we use the f_5 Indicator feature to indicate that the last moved boxes are being moved to the wall. We generate this feature for the last four pushed boxes:

$$f^2_{\text{indicator}}(t,S) = \{f_5(B_p),\ f_5(B_{p-1}),\ f_5(B_{p-2}),\ f_5(B_{p-3})\}$$

Similarly, to be indicative of the *to-groups* task, we use the f_6 Indicator feature to indicate that the last four moved boxes are being moved to within a threshold of a like colored box.

$$f^3_{\text{indicator}}(t,S) = \{f_6(S[t],B_p),\ f_6(S[t],B_{p-1}),\\ f_6(S[t],B_{p-2}),\ f_6(S[t],B_{p-3})\}$$

Finally, to be indicative of the *to-line* task, we look at the past three moved boxes and four moved boxes and indicate if they are within a threshold of a line.

$$f^4_{\text{indicator}}(t,S) = \{f_7(\{B_p,B_{p-1},B_{p-2}\}),\\ f_7(\{B_p,B_{p-1},B_{p-2},B_{p-3}\})\}$$

Thus, the entire set of Indicator features is given by:

$$f_{\text{indicator}}(t,S) = \{f^1_{\text{indicator}}(t,S),\ f^2_{\text{indicator}}(t,S),\\ f^3_{\text{indicator}}(t,S),\ f^4_{\text{indicator}}(t,S)\}$$

IV. Evaluation

A. Introduction

To determine the optimal set of features to use we ran several experiments with different sets of features to test the classification accuracy and speed. Two datasets are used for evaluating our method. The first dataset is generated from a simulated environment where boxes are moved by clicking and dragging box locations. The second dataset is generated by an overhead camera system that monitors and detects positions of real boxes that are physically moved by a person in the workspace (shown in Figure 2). In both cases the positions of the boxes are recorded at a rate of 2Hz. Nine trials were recorded of a user moving the boxes into randomly generated goal positions. One observation sequence consists of nine goals per trial for the simulated dataset, and 6 goals per trial for the physical experiments. The starting configurations of the boxes for each trial are randomly generated. Experiments were conducted using leave-one-out cross-validation where a model is trained on eight observation sequences and tested on the remaining observation sequence. This was repeated so that every observation sequence is used as the testing dataset. For our experiments we used the CRF implementation from [13], which is available as an open source library [1].

The user moves the boxes into one of three previously mentioned goal configurations. Once the boxes are moved

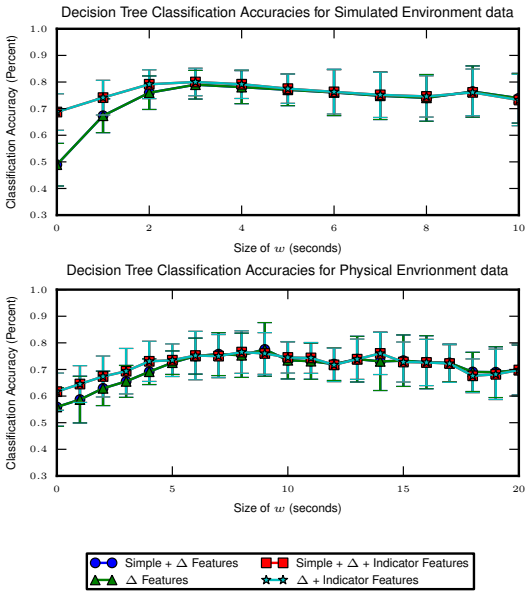


Fig. 4. Decision tree accuracies for the differing feature sets plotted as a function of the Δ size. Top: Simulated environment data. Bottom: Physical environment data. Note that at $w = 0$ the Δ features are disabled.

to the desired configuration for the current goal, a new goal is randomly chosen. In the simulated environment there are 4 boxes of each color for a total of 12 boxes, while in the physical experiments there are 3 boxes of each color, due to size constraints of the workspace.

B. Description of Tests

We trained multiple CRF classifiers using differing sets of features to determine the optimal configuration. First, we test the CRF using only the Simple features; then we test the effects of adding the Δ features as well as the Indicator features. We test the CRF classifiers using differing values of w for the Δ features, which affects the overall time that the Δ feature considers. We also test CRF classifiers without Simple features, using only Δ features as well as Δ and Indicator features (no Simple features). The CRF classifiers are evaluated using two methods: with the full observation sequence, and in an "on-line" method. They are also evaluated using two metrics: average accuracy and Time To Correct Classification (TTCC). The TTCC is defined to be the number of observations required after a goal change to successfully "steadily" classify the human's goal. That is, it is the time from the initial goal change to the first correct classification that remains the correct classification until the goal is finished. For comparison, we also show the performance of a Decision Tree Classifier, as well as human test subjects classifying the observation sequences.

C. Classification Accuracy

1) Simple Features: In Figure 3 we can see the Simple features only as the blue circle line at size $w = 0$. With Simple features alone the CRF does not perform with high accuracy (29% accuracy with on-line classification in the simulated environment). This is likely a result of the Simple features not indicating any transition from one class to another.

2) Change of Features Over Time (Δ Features): The addition of the Δ features provides the information necessary for transitioning from one class to another. As can be seen in Figure 3 the performance of the CRF classifier immediately increases once the window features are added (i.e., the blue line with circle markers increases when $w > 0$.) We can also see that as the w value increases, the performance of the classifier degrades because the information covered in the Δ features is aging and becomes less relevant to the current task.

We also trained CRF classifiers using only the Δ features. The accuracy can be seen as the green line with triangular markers in Figure 3. The CRF model that uses only Δ features has the same shape of accuracy performance as the Simple and Δ features, but performs worse. We can then conclude that despite being calculated from the Simple features, the Δ features cannot replace, but supplement the Simple features.

3) Indicator Features: The red line with square markers in Figure 3 and the cyan line with star markers represent the classification accuracy of a CRF using Simple, Δ and Indicator Features, and a CRF using Δ and Indicator features, respectively. For the data from the simulated environment, both models perform comparably, implying that the Simple features are not needed for high performance. However, for the physical experiment data, the CRF that does not have access to the Simple features is outperformed by the CRF that does have access to the Simple features. This implies that the Indicator features for the physical experiments are not performing as well as in the simulated environment, and therefore susceptible to noise. It is fairly common that the box will appear to stop, but then will start up again soon after. Such occurrences can be caused by many things, such as occlusions by the human as they push the box. We can deal with this problem by increasing the time the box is required to be stopped before marking it as pushed; however as this time increases, the usefulness of the Indicator feature decreases.

4) Decision Tree: For comparison purposes, we show the performance of a Decision Tree for both the Simulated and Physical experiment data in Figure 4. We can see that the Decision Tree relies primarily on the Simple features, since the accuracy performance does not increase when given access to more features. Because the Decision Tree treats every observation as a unique sample, without any temporal information, all classification is done in an "on-line" fashion. We can see that with accuracy as the metric, the Decision Tree can outperform even the best performing CRF model, when the CRF's classification is done on-line. When the CRF has access to the entire data sequence it achieves a higher classification accuracy than the Decision Tree. However, because we are interested in classifying the human's goals as the human performs them, and not afterwards, we are interested in the "on-line" performance. As we will explain in Section IV-D, however, accuracy is not a proper metric

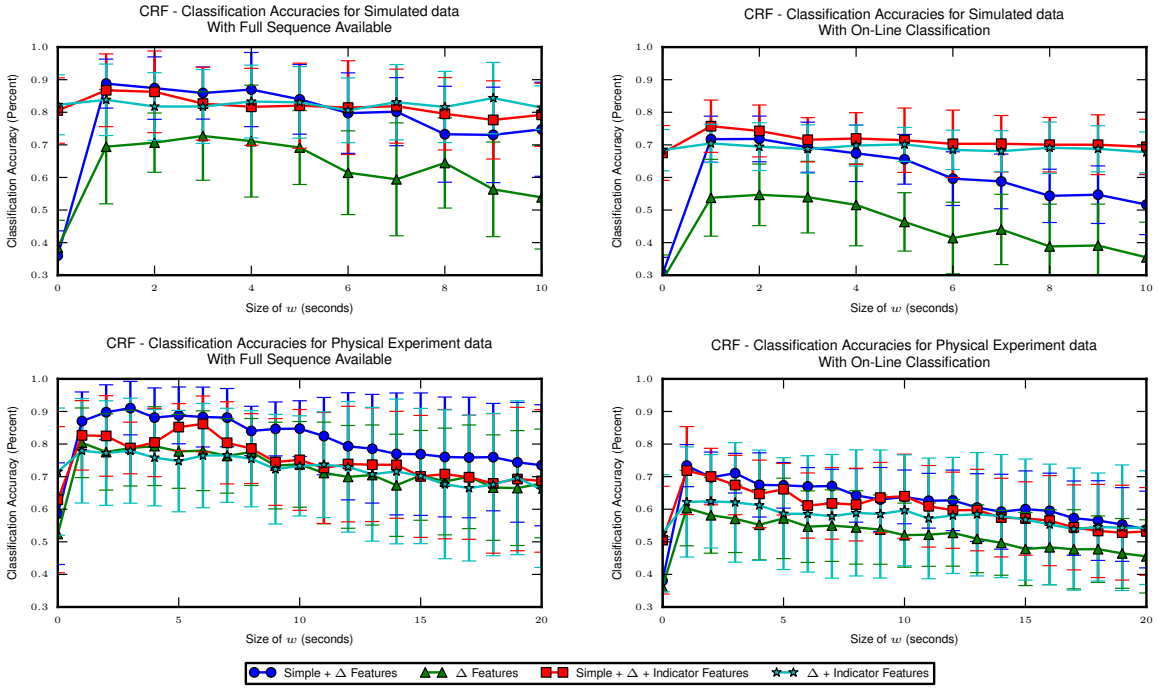


Fig. 3. Accuracies for the different features plotted as a function of the Δ size. Top Left: Simulated environment data with full observation sequence available. Top Right: Simulated environment data with classification performed on-line. Bottom Left: Physical environment data with full observation sequence available. Bottom Right: Physical environment data with classification performed on-line. For on-line classification of the simulated data: when $w < 6$s, Simple + Δ features performs the same as Simple + Δ + Indicator features. However, when $w \geq 6$s the Simple + Δ + Indicator features performs better than the Simple + Δ features CRF with a confidence of 95%. For all cases, Simple + Δ features outperforms Δ features alone with over 95% confidence. Statistically speaking, for the physical environment with on-line classification, the Simple + Δ features performs the same as the Simple + Δ + Indicator features for $w < 3$. Also for $w < 3$ both Simple + Δ and Simple + Δ + Indicator features outperformed Δ + Indicator and Δ only.

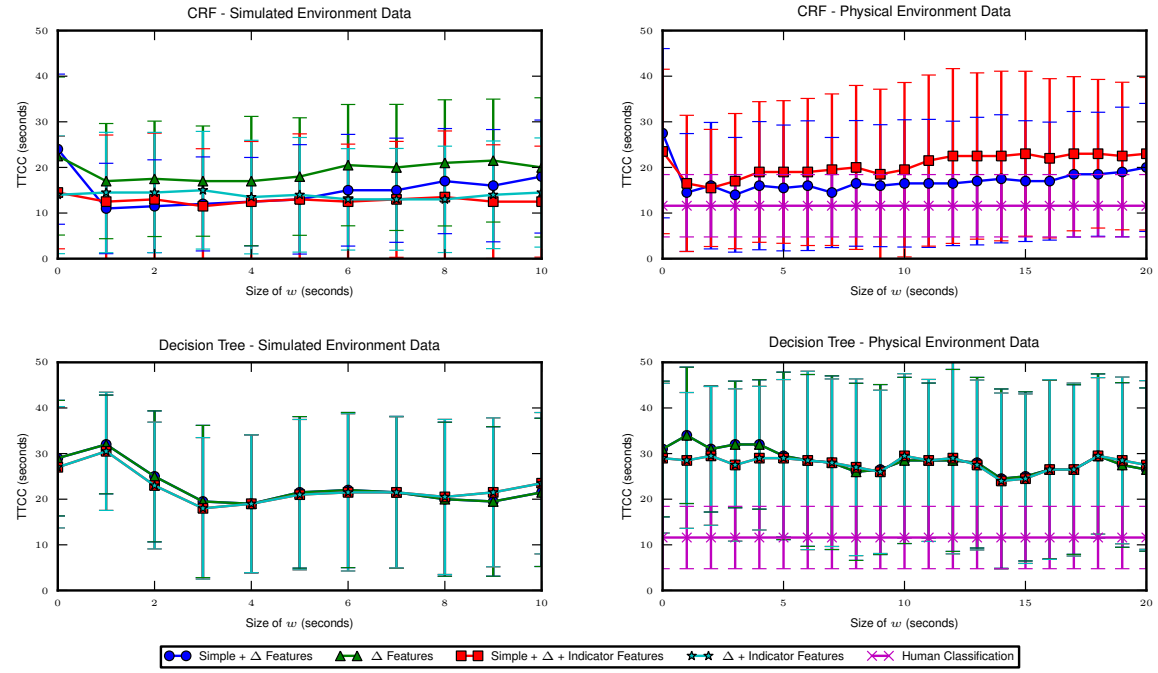


Fig. 5. Average Time To Correct Classification(TTCC) as a function of Δ feature w size using the Conditional Random Fields Classifier (top row) and the Decision Tree classifier (bottom row). Note that the human's classification accuracy is plotted as well. Standard Deviation is given as error bars. Note that for the physical environment data with the CRF (top right) the Δ features and $\Delta+$ Indicator features are omitted for increased clarity. For the CRF with the physical environment data (top right) the results of the Simple + Δ features CRF and Simple + Δ + Indicator are statistically similar with 90% confidence for $w < 11s$ and are statistically different for $w \geq 11$ with 90% confidence and with 95% confidence for $w \geq 12$.

TABLE I

RESULTS OF HUMAN CLASSIFICATION OF THE PHYSICAL EXPERIMENTS, IN TERMS OF ACCURACY AND TIME TO CORRECT CLASSIFICATION (TTCC)

Subject	Avg. Accuracy	Accuracy SD	Avg. TTCC	TTCC SD
A	0.66	0.047	14.5	6.46
B	0.76	0.043	9.5	4.79
C	0.72	0.095	12	8.74
D	0.71	0.047	12.5	10.10
E	0.76	0.043	9.5	4.08

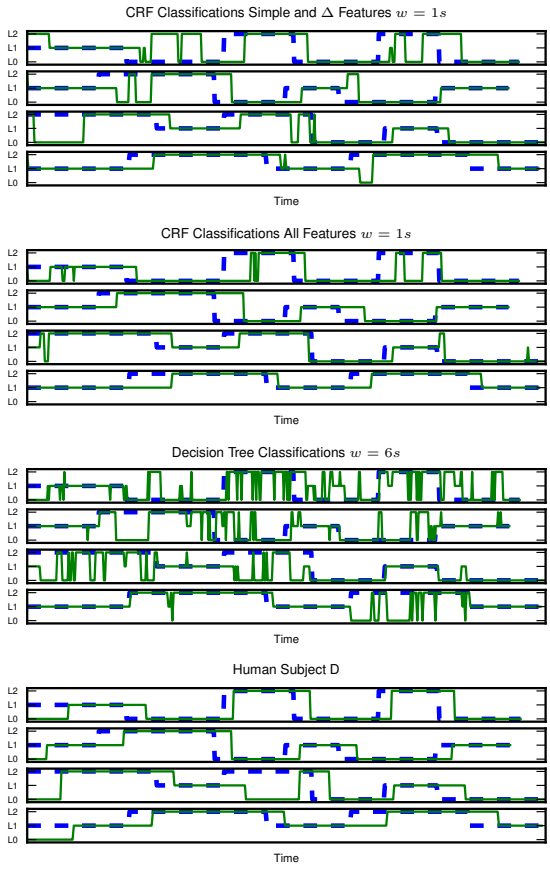


Fig. 6. Timing Diagrams showing ground truth label (Blue Dashed Line) and the classified task (solid green line) for selected classifiers. Each row represents the on-line classification of the given classifier for one test observation sequence.

for our desired objectives, showing how these results are misleading.

5) Human Classification: For comparison, we had human subjects classify the tasks using the same observation sequence that is available to the classifiers. The human subjects were not able to see the human that was moving the boxes, only the result of their actions: the movement of the boxes. The results of the human classification trials are given in Table I. We claim that for this task, the human performance is the best that can be achieved. We perform this test because right after a task change there is simply not enough information to make a classification, and we believe that the human's classification performance shows the best classification that can be made.

D. Time to Correct Classification

For our use, accuracy can be a misleading metric. If we examine the human's classified data in Table I, we can see that the most accurate test subject (both subjects B and E) scored an accuracy of only 76%. In Figure 4 we can see that the Decision Tree classification has a similar accuracy of 78%. However, for our purposes, we are interested in the time it takes to determine the human's goal, not the accuracy of the classifier for the entire sequence. Immediately after the task changes, there is no information regarding the current goal. During this stage of uncertainty, if we wanted to maximize accuracy, we could guess one of the two possible new goals. However, this does not tell us anything about how quickly the classifier can make the correct decision, which is what we are truly interested in.

Since using accuracy as a metric can be misleading, we use the time to correct classification (TTCC) metric to evaluate our models. In Figure 5 we can see that despite the higher accuracy, the Decision Tree has a longer time to correct classification than the CRF models in both the simulated and physical environments. For $1s \geq w \geq 2s$ the classification performance of the CRF with Simple + Δ features performs the same as the CRF with Simple + Δ + Indicator features. For $w \geq 11$ the CRF with Simple + Δ features performs better than the CRF Simple + Δ + Indicator features with 90% confidence. The human classified trials shown in Figure 5 illustrate what is likely the lowest time to correct classification, as it is unlikely that a machine classifier could outperform a human in this task.

We would have expected the CRF with Indicator features to perform better than the CRF without them. In the physical environment we feel that the Indicator features were not performing as well due to the noise and tracking errors from the box tracking system. We feel that because the Indicator features require a memory of pushed boxes, the noise in the tracking system could detect that a box has been pushed prematurely, thus misleading the system. Including information from the human's current location and actions, we feel that the Indicator features can improve accuracy. (This is the subject of future work.)

We can visualize why the CRF models perform better than Decision Trees by referring to Figure 6. This figure shows timing diagrams of how the different classifiers classify the data sequences. The CRF tends to stay with the same label, which allows it to be more robust to noise in the system. The Decision Tree tends to classify accurately on average, but without temporal information, resulting in classifications that are often changing, and not stable. With more data, the CRF becomes more confident in its classification, while the decision tree only classifies using the current observation.

V. TOWARDS PHYSICAL ROBOT IMPLEMENTATION

To initially prove the feasibility of a human and a robot working in the same shared workspace for the defined application, we have implemented a simple collaborative box-pushing controller and human tracker. This system gives the ability for a Pioneer 3DX robot to use its laser scanner,

Fig. 7. Left: The human gets ready to push the box on the left side. Center: The robot moves into position to help push on the right side of the box. Right: The human and the robot work together to push the box.

plus a robot-built map of the environment, to recognize the location of a human-selected box in the environment. In these experiments, the selected box is the one that the human is currently pushing. The idea is that the box is too long for the human to push alone, and requires robot assistance. For rapid implementation, we made use of the laser-based public domain software package [20] that performs fast line, arc/circle, and human leg detection. Figure 7 shows some snapshots of these preliminary physical robot demonstrations. In these experiments, the robot has a map of its environment; it uses its laser to detect a box and the legs of a human in its field of view. (The robot uses the map to eliminate walls as possible boxes.) When the human is detected, the robot determines his or her position along the box. From this information, the robot decides which end of the box it should move to, and then moves appropriately. Videos of this interaction are available at `http://www.cs.utk.edu/dilab/`. The system presented in this paper represents an environmental monitoring system. Our overall system will use this environmental monitoring system with supplemental information from a human activity recognition system as input to an action-selection process that the robot will use to determine the best box to push, and what goal position to place the box. Together, these individual systems form an overall system to enable robots to work with a human on a shared workspace task, without relying on explicit communication.

VI. Conclusions and Future Work

This paper has presented a concept for peer-to-peer human robot teaming using implicit coordination. This concept requires an environmental monitor to determine the human's intent, in the form of a desired goal state. We presented a design for the environmental monitor using Conditional Random Fields. We have concluded that in the simulation environment the Indicator features in addition to the Δ and Simple features improve performance. However, in the physical experiments the Indicator features do not appear to improve performance. We determined that a small window size is best for both accuracy and TTCC. We hypothesize that this is because the Indicator features rely on the history of the boxes being pushed, which is not as robust in the physical environment due to noise. For future work we intend on looking at the actions the human is taking directly to determine what box they are pushing, which we believe will increase accuracy and speed. We also plan on incorporating an action selection process to enable the robot to decide what action to take in order to help the human accomplish the task using implicit coordination.

VII. Acknowledgements

The authors gratefully acknowledge the support of Lockheed Martin Advanced Technology Laboratories in this research. This material is based in part upon work supported by the National Science Foundation under Grant No. 0812117.

References

[1] Hcrf library (including crf and ldcrf). `http://sourceforge.net/projects/hcrf/`.

[2] M.E. Bratman. Shared cooperative activity. *The Philosophical Review*, 101(2):327–341, 1992.

[3] C. Bregler. Learning and recognizing human dynamics in video sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1997.

[4] R. T. Collins, A. J. Lipton, H. Fujiyoshi, and T. Kanade. Algorithms for cooperative multisensor surveillance. *Proceedings of the IEEE*, 89(10):1456–1477, 2001.

[5] M. Desai and H. A. Yanco. Blending human and robot inputs for sliding scale autonomy. In *IEEE International Workshop on Robot and Human Interactive Communication*, volume 537–542, 2005.

[6] M. B. Dias, B. Kannan, B. Browning, E. Jones, B. Argall, M. F. Dias, M. B. Zinck, M. Veloso, and A. Stentz. Sliding autonomy for peer-to-peer human-robot teams. Technical Report CMU-RI-TR-08-16‘, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, April 2008.

[7] G. Hoffman and C. Breazeal. Collaboration in human-robot teams. In *Proceedings of the 1st AIAA Intelligent Systems Technical Conference*, 2004.

[8] G. Hoffman and C. Breazeal. Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team. In *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, page 8. ACM, 2007.

[9] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Machine Learning-International Workshop Then Conference*, pages 282–289, 2001.

[10] L. Liao, D. Fox, and H. Kautz. Extracting places and activities from gps traces using hierarchical conditional random fields. *The International Journal of Robotics Research*, 26(1):119, 2007.

[11] D.C. Liu and J. Nocedal. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.

[12] J.L. Marble, D.J. Bruemmer, D.A. Few, and D.D. Dudenhoeffer. Evaluation of supervisory vs. peer-peer interaction for human-robot teams. *Proceedings of the Annual Hawai International Conference on System Sciences*, 2004.

[13] L.P. Morency, A. Quattoni, T. Darrell, and C. MIT. Latent-dynamic discriminative models for continuous gesture recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07*, pages 1–8, 2007.

[14] K.B. Reed and M.A. Peshkin. Physical collaboration of human-human and human-robot teams. *Haptics, IEEE Transactions on*, 1(2):108 – 120, july-dec. 2008.

[15] Paul Scerri, Katia Sycara, and M. Tambe. Adjustable autonomy in the context of coordination. In *AIAA 3rd "Unmanned Unlimited" Technical Conference, Workshop and Exhibit*, 2004. Invited Paper.

[16] C. Sutton and A. McCallum. An introduction to conditional random fields for relational learning. In L. Getoor and B. Taskar, editors, *An Introduction to Statistical Relational Learning*. MIT Press, 2007.

[17] F. Tang and L. E. Parker. Layering ASyMTRe-D with task allocation for multi-robot tasks. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2007.

[18] D. L. Vail, J. D. Lafferty, and M. M. Veloso. Feature selection in conditional random fields for activity recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2007.

[19] H. Wallach. Efficient training of conditional random fields. In *Proc. 6th Annual CLUK Research Colloquium*, volume 112. Citeseer, 2002.

[20] J. Xavier, M. Pacheco, D. Castro, A. Ruano, and U. Nunes. Fast line, arc/circle and leg detection from laser scan data in a player driver. In *IEEE International Conference on Robotics and Automation*, 2005.